

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A method of using a computational effort invested in a proof of work (POW), the method executable in one or more processors in communication with one or more memory devices having embodied therein stored programs for performing the method, comprising: distributing a computational task among a plurality of entities for execution within a specified interval of time as a POW; receiving a the POW relating to said task from one of said plurality of entities; ~~and~~ using said POW to accomplish said task, and distribution of the POW as a POW.

2. (Original) The method of claim 1 further comprising using said POW to accomplish a security goal.

3. (Original) The method of claim 1 , wherein distributing said task among a plurality of entities includes partitioning said task into a plurality of sub-computational tasks and distributing each one of said plurality of sub-computational tasks to a respective one of said plurality of entities;

4. (Currently Amended) The method of claim ~~1~~ 2 wherein said security goal involves restricting resource access by said one of said plurality of entities.

5. (Currently Amended) A method of using a computational effort invested in a proof of work (POW), the method executable in one or more processors in communication with one or more memory devices having embodied therein stored programs for performing the method, comprising: partitioning a minting operation into a plurality of sub-computational tasks; distributing one of said plurality of sub-computational tasks to one of a plurality of entities; receiving a POW from said one of said plurality of entities; ~~and~~ using said POW to accomplish said minting operation , and distribution of the POW as a POW

6. (Original) The method of claim 5 further comprising using said POW to accomplish a security goal.

7. (Original) The method of claim 5 wherein said minting operation includes identifying valid solutions that hash to a predetermined image and wherein said POW represents a valid solution.

8. (Original) The method of claim 6 wherein said predetermined image comprises a range of images.

9. (Original) The method of claim 8 wherein all images within said range of images have a predetermined number of least significant bits in common.

10. (Original) The method of claim 5 wherein each of said sub-tasks comprises searching a different solution search space for valid solutions.

11. (Original) The method of claim 6 wherein said security goal involves restricting resource access.

12. (Original) The method of claim 7 further comprising verifying said valid solution by determining whether said valid solution represented by said POW hashes to said predetermined image.

13. (Currently Amended) A method of using a computational effort invested in a proof of work (POW) , the method executable in one or more processors in communication with one or more memory devices having embodied therein stored programs for performing the method, comprising: distributing a minting operation among a plurality of entities in a manner that maintains privacy in said minting operation; receiving a POW from said one of said plurality of entities relating to said minting operation; ~~and~~ using said POW to accomplish said minting operation , and distribution of the POW as a POW.

14. (Original) The method of claim 13 further comprising using said POW to accomplish a security goal.

15. (Original) The method of claim 13 wherein said minting operation comprises using a hash function to identify a predetermined number of valid solutions that hash to a target value and wherein said POW represents a valid solution.

16. (Original) The method of claim 15 wherein said predetermined number of valid solutions comprise a coin.

17. (Original) The method of claim 15 wherein said predetermined number of valid solutions hash to a portion of said target value.

18. (Original) The method of claim 13 wherein said distributing includes instructing each of said plurality of entities to search within a different search space for valid solutions.

19. (Original) The method of claim 15 wherein said privacy is maintained in said minting operation by keying said hash function with a secret value.

20. (Original) The method of claim 19 wherein said secret value includes a portion specific to a coin.

21. (Original) The method of claim 20 wherein said secret value includes a portion specific to a period of said coin's validity.

22. (Original) The method of claim 19 wherein said hash is of a concatenation of a solution and a value generated using said secret value.

23. (Original) The method of claim 13 further comprising verifying said POW.

24. (Currently Amended) A method of using a computational effort invested in a proof of work (POW), the method executable in one or more processors in communication with one or more memory devices having embodied therein stored programs for performing the method, comprising:

generating a computational task for a certain amount of intense computation in a specified period of time as a POW to accomplish a separate, useful and verifiable correct computation;

distributing the computational task for execution among a plurality of server entities

receiving a POW relating to said task from one of said plurality of said server entities;

and

using said POW to verify and accomplish said computational task, and distribution of the POW as a POW.

25. (Previously Presented) The method of claim 24 wherein the proof of work POW is  $(w, p)$ -hard if prover P with memory resources bounded by  $m$  performs an average, over all coin flips by P and V, of at most  $w$  steps of computation in the time interval  $[t_s, t_c]$ , and the verifier V accepts with probability at most  $p + o\left(\frac{m}{\text{poly}(l)}\right)$ , where  $l$  is a security parameter;  $t_s$  is start time and  $t_c$  is complete time.

26. (Previously Presented) The method of claim 24 wherein a proof of work POW is  $(w, p, m)$ -feasible if there exists a prover P with memory resources bounded by  $m$  such that with an average of  $w$  steps of computation in the time interval  $[t_s, t_c]$ , the prover can cause the verifier V to accept with probability at least  $p$ .

27. (Previously Presented) The method of claim 24 wherein a proof of work POW is sound, if, for some  $w$ , POW is  $(w, 1, \text{poly}(l))$ -feasible, where  $l$  is a security parameter.

28. (Previously Presented) The method of claim 24 wherein a POW may be regarded as efficient if the verifier performs substantially less computation than the prover.